**Linux Notes Basics.   
  
What is Linux?**  
Linux is an open-source operating system (OS) based on the Unix architecture. It is case sensitive It serves as the core software that manages hardware resources, provides a user interface, and offers a platform for running applications. It is widely known for its flexibility, stability, and security, making it popular in various environments like servers, desktops, embedded systems, and more.

**1. Open Source:** The source code is freely available and can be modified by anyone.

**2. Multi-User and Multi-Tasking:** Supports multiple users and can run several processes simultaneously.

**3. High Stability and Security:** Known for reliability and strong security features.

**4. Customization:** Users can tailor the system to their specific needs.  
  
**Categories of Linux:**

Linux distributions can be broadly categorized into different types based on their use cases, target audience, and the functionalities they offer. Here are the main categories:

**1. Minimal OS**

**- Purpose:** Provides only essential components for minimal resource usage. Often used in servers or lightweight environments.

**- Examples:** Arch Linux, Alpine Linux, Gentoo, Linux From Scratch.

**2. Desktop OS**

**- Purpose:** Designed for everyday users, offering a user-friendly interface and pre-installed applications.

**- Examples:** Ubuntu, Fedora, Linux Mint, Zorin OS.

**3. Server OS**

**- Purpose:** Tailored for server environments with a focus on stability, security, and performance.

**- Examples:** CentOS, Red Hat Enterprise Linux (RHEL), Debian, SUSE Linux Enterprise Server (SLES).

**4. Enterprise OS**

**- Purpose:** Offers extended support and enterprise-grade features, suitable for businesses and large organizations.

**- Examples:** Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise, Oracle Linux.

**5. Specialized OS**

**- Purpose:** Developed for specific use cases such as multimedia production, scientific computing, or education.

**- Examples:** Ubuntu Studio (multimedia), Scientific Linux (research), Edubuntu (education).  
 **6. Embedded OS 7. Security-Focused OS ..etc**

**1. Linux OS Installation on VMware**

- Installing Linux on VMware involves setting up virtual partitions. Efficient partitioning improves system performance and organization. Common partitions include:

**- `/` (root):** The primary partition containing all system files.

**- `/home`:** Stores user files and settings.

**- `/var`:** Stores variable data like logs.

**- `/boot`:** Contains bootloader files.

**- `swap`:** A swap partition acts as virtual memory.

**2. What is `/` (root)?**

- The **`/`** directory, called root, is the base of the Linux filesystem. All other directories **(like `/home`, `/var`, and `/usr`)** branch out from the root directory. It’s where the core system files and directories reside.

**3. How to Start Terminal?**

- You can start a terminal in different ways:

- Press **Alt + F2** and type `gnome-terminal`.

- Use **FN + Alt + F2** on laptops.

- **Right-click** on the desktop and select **"Open in Terminal".**

- Use shortcut keys like **Ctrl + Alt + T** (for Ubuntu-based distros).

**4. Different Types of Terminals in Linux**

- Linux offers various terminal types:

**- gnome-terminal:** Common in GNOME desktop environments.

**- xterm:** A simple and lightweight terminal.

**- konsole:** Used in KDE environments.

**- tilix:** A tiling terminal emulator.

**5. Commands to Check Partitions after Installation**

- After installing Linux, you can check the partitions using these commands:

**- `df`:** Displays disk usage.

**- `df -h`:** Displays disk usage in a human-readable format.

**- `lsblk`:** Lists all block devices.

**- `fdisk -l`:** Lists all partitions and details.

**5. Remote Software Login**

Remote login allows users to access and control a computer from a different location. Here are some commonly used tools and methods for remote login:

**1. PuTTY:**

- A widely used SSH and Telnet client for Windows. It allows secure remote connections to Linux servers using SSH.

**2. MobaXterm:**

- An advanced terminal that supports SSH, SFTP, and various network protocols. It also includes an X11 server for remote desktop functionality.

**3. Git BASH:**

- A Windows-based terminal that provides a Bash emulation environment. It supports SSH for connecting to remote servers and is often used by developers working with Git repositories.

**4. SSH (Secure Shell):**

- A secure protocol used to log in and execute commands on a remote Linux/Unix server. Commonly used command: `ssh username@hostname`.

**5. SMTP (Simple Mail Transfer Protocol):**

- Although primarily used for email transmission, SMTP can also be configured for remote login in certain scenarios involving communication between systems.

These tools and protocols offer versatile options for securely accessing and managing remote systems.

**7. Linux System Hierarchy**

- The Linux filesystem follows a hierarchical structure:

**- `/bin`:** Essential command binaries.

**- `/etc`:** Configuration files.

**- `/var`:** Variable data like logs.

**- `/usr`:** User utilities and applications.

**- `/home`:** User directories.

**8. Basic Commands**

- Here are some fundamental Linux commands:

**- `pwd`:** Shows the current directory.

**- `cd ..`:** Moves up one directory level.

**- `ls -l`:** Lists directory contents.  
**- `hostname`:** Displays the hostname.

**- `hostname -i`:** Shows the IP address.

**- `ifconfig` or `ip addr`:** Displays network interfaces.

**- `touch`:** Creates an empty file.

**- `mkdir`:** Creates a directory.

**- `cp`:** Copies files or directories.

**- `mv`:** Moves or renames files.

**- `history`:** Shows command history.

**- `lsblk`:** Lists block devices.

**- `free -h`:** Shows memory usage.

**- `lscpu`:** Displays CPU architecture information.

**- `cat`, `tail`, `head`:** View file contents.

**- `telnet`:** For testing connections to servers.

**- `nc -zv`:** Network connectivity testing.  
 **- `top`:** Shows real-time system stats including processes, CPU, and memory.

**- `ps -ef | grep <app>`:** Lists all processes and filters by a specific application.

**- `kill`:** Sends a signal to terminate a process. Use `**kill PID**` or `**kill -9 PID**` to force terminate.

**- `uptime`:** Shows system uptime, current time, and load averages.  
  
**- `tar`:** Archives files into a single file or extracts files from an archive.

**- `gzip`:** Compresses files using the gzip format.

**- `gunzip`:** Decompresses files compressed with gzip.

**- `ping`:** Tests network connectivity by sending ICMP echo requests.

**- `rm`:** Removes files or directories.

**- `chmod`:** Changes file permissions.

**- `chown`:** Changes file ownership.

**- `wget`:** Downloads files from the web.

**- `yum`:** Manages packages in RPM-based distributions (e.g., CentOS).

**- `whoami`:** Displays the current logged-in user.

**- `sudo`:** Executes a command with superuser privileges.

**- `su`:** Switches to another user account.

**- `passwd`:** Changes user passwords.

**- `useradd`:** Adds a new user account.

**- `whereis`:** Locates binaries, source files, and man pages.

**- `whatis`:** Displays a one-line description of a command.

**- `man`:** Shows the manual page for a command.

**- `wc`:** Counts lines, words, and characters in a file.

**- `exit`:** Exits the current shell session.  
When we use a hyphen (**`-`**) followed by options or flags in commands like **`df -`, `ls -`, or `ps -`**, these options are referred to as **command-line flags** or **command options**. They modify the behaviour of the command and determine how it outputs information.

**For example:**

- In `**ls -ltr`,** `**-ltr**` are the flags/options that control the format, sorting, and order of the listing.

- In `**df -h**`, `**-h**` is a flag that tells the `**df**` command to display disk usage in a human-readable format.

These flags are typically:

1. **Single-character flags** (like **`-l`, `-h`, `-r`**).

2. Sometimes combined together (like **`-ltr` or `-aux`).**

You can refer to them as **"options," "flags," or "parameters."**

**9. Symlink and Hard Link: Advantages and Disadvantages**

**- Symlink (Symbolic Link):**

- Creates a reference to another file.

- Can link across different filesystems.

- Breaks if the target file is deleted.

- Command: `**ln -s target linkname**`

**- Hard Link:**

- Creates a direct pointer to the data on disk.

- Remains valid even if the original file is deleted.

- Cannot link across filesystems.

- Command: `**ln target linkname**`

**10. Difference between `cp` and `rsync`: Advantages and Disadvantages**

**- `cp`:**

- Simple copy command.

- Useful for local file copying.

- No built-in sync or incremental options.

**- `rsync`:**

- Designed for backups and synchronizations.

- Supports incremental copying and remote sync.

- Efficient over networks.

**11. Vi and Vim Basics: Advantages and Disadvantages**

**- Vi:**

- A lightweight text editor available by default on most Linux systems.

- Minimal features.

**- Vim (Vi Improved):**

- An enhanced version with additional features like syntax highlighting.

- Supports plugins and extended configurations.

**- Basic Commands:**

**- Insert Mode:** Press **`i`** to start editing.

**- Command Mode:** Press **`Esc`** to switch back.

**- Save and Exit:** **`:wq`.**

**- Quit without saving:** **`:q!`.**- Master basic commands in Vi/Vim for efficient text editing.

**Navigation Commands**

**- `Up`:** Move the cursor up one line **`k`.**

**- `Down`:** Move the cursor down one line **`j`.**

**- `Page End`:** Move to the end of the file **`G`.**

**- `Page Top`:** Move to the beginning of the file **`gg`.**

**Editing Commands**

**- `Copy`:** Copy text. In visual mode, use **`v`** to select text, then **`y`** to yank (copy).

**- `Paste`:** Paste text. Use **`p`** to paste after the cursor or **`P`** to paste before the cursor.

**- `Delete Line`:** Delete the current line **`dd`.**

**- `Delete All`:** Delete all lines in the file (`:%d`).

**- `Select Word`:** Select a word. Place the cursor on the word and use `viw` to select the word in visual mode.

**- `Select Line`:** Select the entire line. Use `V` to enter visual line mode.

**- `Select All`:** Select all text. Use `ggVG` to select all text from the top to the bottom.

**Replacement Commands**

**- `Replace Word`:** Replace a specific word. Place the cursor on the word and use `cw` to change the word. After entering the new word, press `Esc`.

**12. File Permissions in Linux**

File permissions in Linux determine who can read, write, or execute a file. Each file has three types of users:

**1. Owner:** The user who owns the file.

**2. Group:** A group of users who have specific permissions.

**3. Others:** All other users.

**Permission Values:**

Each permission is represented by a number:

**- Read (r): `4`**

**- Write (w): `2`**

**- Execute (x): `1`  
Examples of Permission Combinations:**

**- 7 (rwx):** `Read (4)` + `Write (2)` + `Execute (1)` = `7` (Full permission: read, write, execute).

**- 6 (rw-):** `Read (4)` + `Write (2)` = `6` (Read and write only).

**- 5 (r-x):** `Read (4)` + `Execute (1)` = `5` (Read and execute only).

**- 0 (---):** No permissions.

**Example Permission Setting:**

**- 7 (owner):** `rwx` (Read, write, and execute).

**- 5 (group):** `r-x` (Read and execute).

**- 5 (others):** `r-x` (Read and execute).

This example would be represented as **`755`,** and it grants:

**- Owner:** Full permissions (**rwx**).

**- Group:** Read and execute (**r-x**).

**- Others:** Read and execute (**r-x**).

**Changing Permissions:**

- Use the **`chmod`** command to set file permissions:

**chmod 755 filename**

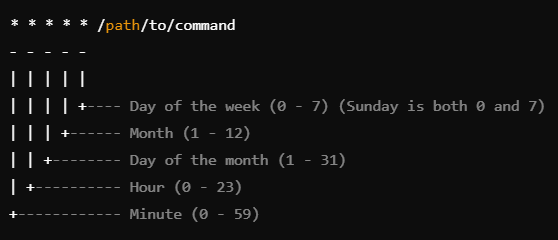
This command sets the permissions according to the example above.  
  
**Crontab:**  
  
**What is Cron?**

* **Definition**: cron is a time-based job scheduler in Unix-like operating systems. It allows users to run scripts, commands, or programs at scheduled intervals.

**What is Crontab?**

* **Definition**: crontab (short for "cron table") is a file that contains a list of cron jobs. Each line in this file represents a job and its schedule.

**Crontab Syntax**

Each line in a crontab file follows this general format:  


To schedule a script to run daily at 8 AM using cron.  
  
  
  
  
**Crontab.guru:**https://crontab.guru/ is a useful online tool for creating and understanding cron expressions. It provides a simple interface where you can:

**- Generate Cron Expressions:** Input your desired schedule and get the corresponding cron expression.

**- Understand Cron Expressions:** See a human-readable description of what a given cron expression does.

**- Test Cron Expressions:** Enter a cron expression to see when it will next run.

It's a handy tool for anyone who needs to work with cron jobs but wants to ensure their cron syntax is correct and understandable.